**APP.JSX :**

**import React from 'react';**

**import { BrowserRouter as Router, Routes, Route } from 'react-router-dom';**

**import Sidebar from './Components/Sidebar';**

**import Shoes from './Components/Pages/Shoes';**

**import Addshoes from './Components/Pages/Addshoes';**

**import Customers from './Components/Pages/Customers';**

**import Addcustomers from './Components/Pages/Addcustomers';**

**import Sales from './Components/Pages/Sales';**

**import Addsales from './Components/Pages/Addsales';**

**import Suppliers from './Components/Pages/Suppliers';**

**import Addsuppliers from './Components/Pages/Addsuppliers';**

**import Purchases from './Components/Pages/Purchases';**

**import Home from './Components/Pages/Home';**

**import Navbar from './Components/Navbar';**

**import Addpurchases from './Components/Pages/Addpurchases';**

**const App = () => {**

**return (**

**<div className="">**

**<Navbar />**

**<Router>**

**<div className="h-full">**

**<Sidebar />**

**<div className='sm:ml-64 '>**

**<Routes >**

**<Route path="/" element={<Home />} />**

**<Route path="/shoes" element={<Shoes />} />**

**<Route path="/add\_shoes" element={<Addshoes />} />**

**<Route path="/customers" element={<Customers />} />**

**<Route path="/add\_customers" element={<Addcustomers />} />**

**<Route path="/sales" element={<Sales />} />**

**<Route path="/add\_sales" element={<Addsales />} />**

**<Route path="/suppliers" element={<Suppliers />} />**

**<Route path="/add\_suppliers" element={<Addsuppliers />} />**

**<Route path="/purchases" element={<Purchases />} />**

**<Route path="/add\_purchases" element={<Addpurchases />} />**

**</Routes>**

**</div>**

**</div>**

**</Router>**

**</div>**

**);**

**}**

**export default App;**

**MAIN.JSX :**

**import React from 'react'**

**import ReactDOM from 'react-dom/client'**

**import App from './App.jsx'**

**import './index.css'**

**ReactDOM.createRoot(document.getElementById('root')).render(**

**<React.StrictMode>**

**<App />**

**</React.StrictMode>,**

**)**

**NAVBAR.JSX :**

**import React from 'react'**

**const Navbar = () => {**

**return (**

**<div className="bg-gray-950 text-2xl col p-0 m-0 font-bold shadow-md">**

**<div className="p-2 shadow-md">**

**<h4 className='text-white flex justify-center'>SHOES MANAGEMENT SYSTEM</h4>**

**</div>**

**</div>**

**)**

**}**

**export default Navbar**

**SIDEBAR.JSX :**

**import React from 'react';**

**import { FaHome, FaShoppingBag, FaRupeeSign } from 'react-icons/fa';**

**import { FaChild, FaHandshake } from 'react-icons/fa6';**

**import { GiConverseShoe } from 'react-icons/gi';**

**import { Link } from 'react-router-dom';**

**const Sidebar = () => {**

**return (**

**<>**

**<div className='border-black inline absolute w-64 bg-gray-300 h-full shadow-lg'>**

**<div className='h-16 flex items-center justify-center'>**

**<h1 className="text-2xl font-bold">TABLES</h1>**

**</div>**

**<hr />**

**<ul className="font-bold border-black">**

**<Link to="/" className="px-3">**

**<li className="text-xl rounded hover:shadow hover:bg-black hover:text-white py-5 px-7 cursor-pointer">**

**<FaHome className='inline-block w-6 h-6 mr-2 -mt-2'></FaHome>**

**Home**

**</li>**

**</Link>**

**<Link to="/shoes" className="px-3">**

**<li className="text-xl rounded hover:shadow hover:-white hover:bg-black hover:text-white py-5 px-7 cursor-pointer">**

**<GiConverseShoe className='inline-block w-6 h-6 mr-2 -mt-2'></GiConverseShoe>**

**Shoes**

**</li>**

**</Link>**

**<Link to="/customers" className="px-3">**

**<li className="text-xl rounded hover:shadow hover:bg-black hover:text-white py-5 px-7 cursor-pointer">**

**<FaChild className='inline-block w-6 h-6 mr-2 -mt-2'></FaChild>**

**Customers**

**</li>**

**</Link>**

**<Link to="/sales" className="px-3">**

**<li className="text-xl rounded hover:shadow hover:bg-black hover:text-white py-5 px-7 cursor-pointer">**

**<FaRupeeSign className='inline-block w-6 h-6 mr-2 -mt-2'></FaRupeeSign>**

**Sales**

**</li>**

**</Link>**

**<Link to="/suppliers" className="px-3">**

**<li className="text-xl rounded hover:shadow hover:bg-black hover:text-white py-5 px-7 cursor-pointer">**

**<FaHandshake className='inline-block w-6 h-6 mr-2 -mt-2'></FaHandshake>**

**Suppliers**

**</li>**

**</Link>**

**<Link to="/purchases" className="px-3">**

**<li className="text-xl rounded hover:shadow hover:bg-black hover:text-white py-5 px-7 cursor-pointer">**

**<FaShoppingBag className='inline-block w-6 h-6 mr-2 -mt-2'></FaShoppingBag>**

**Purchases**

**</li>**

**</Link>**

**</ul>**

**</div>**

**</>**

**)**

**}**

**export default Sidebar**

**GRAPH & CHART :**

**import React, { useEffect } from 'react';**

**import Chart from 'chart.js/auto'; // Import Chart.js library**

**const CombinedChart = () => {**

**useEffect(() => {**

**// Get the canvas element**

**var canvas = document.getElementById('combinedChart');**

**if (!canvas) return; // Ensure the canvas element exists**

**var ctx = canvas.getContext('2d'); // Define the context**

**if (!ctx) return; // Ensure the context is valid**

**// Check if a chart instance already exists**

**if (window.combinedChart instanceof Chart) {**

**// If a chart instance exists, destroy it before creating a new one**

**window.combinedChart.destroy();**

**}**

**// Dataset of shoe prices**

**var barData = [14000, 20000, 10000, 4000, 6000];**

**var lineData = [16000, 20000, 6000, 4000, 10000];**

**// Create the combined chart**

**window.combinedChart = new Chart(ctx, {**

**type: 'bar',**

**data: {**

**labels: ['Nike', 'Air Jordans', 'Adidas', 'Puma', 'Woodland'],**

**datasets: [{**

**label: 'Customer Purchases (Bar)',**

**data: barData,**

**backgroundColor: 'rgba(112, 112, 160, 0.65)', // Bar color**

**borderWidth: 1**

**}, {**

**label: 'Sales (Line)',**

**data: lineData,**

**fill: false,**

**borderColor: 'rgba(0, 71, 144, 1)', // Line color**

**borderWidth: 2,**

**type: 'line'**

**}]**

**},**

**options: {**

**scales: {**

**yAxes: [{**

**ticks: {**

**beginAtZero: true**

**}**

**}]**

**}**

**}**

**});**

**}, []); // Empty dependency array ensures this effect runs only once after initial render**

**return <canvas id="combinedChart" width="1000" height="400" className='mb-0'></canvas>;**

**};**

**export default CombinedChart;**

**HOME :**

**import React from 'react'**

**import CombinedChart from '../Combinedchart'**

**import { RiMoneyRupeeCircleLine } from 'react-icons/ri'**

**import { TbDeviceAnalytics } from 'react-icons/tb'**

**const Home = () => {**

**return (**

**<>**

**<div className='mb-0 flex flex-col items-center'>**

**<div className="mt-19 flex justify-center items-center h-290 w-540">**

**<CombinedChart />**

**</div>**

**<div className='text-3xl text-white font-bold flex flex-row space-x-20 justify-between'>**

**<div className="relative hover:shadow-xl">**

**<div className="bg-gradient-to-r from-slate-900 to-gray-700 space-y-8 mb-0 border rounded-lg p-6 h-64 w-96 border-inherit cursor-pointer transition-transform duration-300 hover:scale-110 ">**

**<RiMoneyRupeeCircleLine />**

**<div className="text-4xl text-white font-bold mb-2">Purchases</div>**

**<div className="absolute bottom-20 left-0 w-full border-b-2 border-white"></div> {/\* Line underneath \*/}**

**<div className="absolute bottom-6 left-6 text-white">Total Amount: 46,000</div> {/\* Total amount \*/}**

**</div>**

**</div>**

**<div className="relative">**

**<div className="bg-gradient-to-r from-slate-900 to-gray-700 space-y-8 mb-0 border rounded-lg shadow-lg p-6 h-64 w-96 border-inherit cursor-pointer transition-transform duration-300 hover:scale-110 ">**

**<TbDeviceAnalytics />**

**<div className="text-4xl text-white font-bold mb-2">Sales</div>**

**<div className="absolute bottom-20 left-0 w-full border-b-2 border-white"></div> {/\* Line underneath \*/}**

**<div className="absolute bottom-6 left-6 text-white">Total Amount: 50,000</div> {/\* Total amount \*/}**

**</div>**

**</div>**

**</div>**

**</div>**

**</>**

**)**

**}**

**export default Home**

**SHOES.JSX :**

**import React, { useRef } from 'react'**

**import { Link } from 'react-router-dom'**

**import Shoetable from './Shoetable'**

**const Shoes = () => {**

**const onButtonClick = () => {**

**const pdfUrl = "C:/Projects/Shoes website/Web Docs/sample.pdf";**

**const link = document.createElement("a");**

**link.href = pdfUrl;**

**link.download = "shoes.pdf"; // specify the filename**

**link.target = "\_blank";**

**document.body.appendChild(link);**

**link.click();**

**document.body.removeChild(link);**

**};**

**return (**

**<div className="px-5 w-full h-full flex flex-col justify-center items-center">**

**<div className="text-xl font-bold flex justify-center h-full">**

**<h3 className='text-2xl mt-3 space-y-5'>Shoes Table</h3>**

**</div>**

**<Link to="/add\_shoes" className="hover:bg-gray-800 bg-gray-300 text-black hover:text-white font-bold py-2 px-4 rounded mt-4 inline-block">**

**Add to Table**

**</Link>**

**<div></div>**

**<Shoetable />**

**</div>**

**)**

**}**

**export default Shoes**

**SHOES TABLE :**

**import axios from 'axios';**

**import React, { useEffect, useState} from 'react';**

**const Shoetable = () => {**

**const [shoes, setShoes] = useState([]);**

**const [currentPage, setCurrentPage] = useState(1);**

**const [shoesPerPage] = useState(5);**

**const fetchShoes = async () => {**

**try {**

**const res = await axios.get("http://localhost:3000/shoes");**

**setShoes(res.data.Result);**

**} catch (error) {**

**console.error("Error fetching shoes:", error);**

**}**

**};**

**useEffect(() => {**

**fetchShoes();**

**}, []);**

**// Get current shoes for the current page**

**const currentShoes = shoes.slice((currentPage - 1) \* shoesPerPage, currentPage \* shoesPerPage);**

**// Change page**

**const paginate = (pageNumber) => setCurrentPage(pageNumber);**

**return (**

**<div className="overflow-x-hidden w-full mt-10">**

**<h2 className="text-2xl font-semibold mb-4 text-center">List of Shoe Orders</h2>**

**<table className="table-auto w-full border-collapse">**

**<thead>**

**<tr>**

**<th className="px-6 py-3 bg-gray-200 text-gray-700 font-semibold uppercase border border-gray-300">Shoe ID</th>**

**<th className="px-6 py-3 bg-gray-200 text-gray-700 font-semibold uppercase border border-gray-300">Brand</th>**

**<th className="px-6 py-3 bg-gray-200 text-gray-700 font-semibold uppercase border border-gray-300">Size</th>**

**<th className="px-6 py-3 bg-gray-200 text-gray-700 font-semibold uppercase border border-gray-300">Color</th>**

**<th className="px-6 py-3 bg-gray-200 text-gray-700 font-semibold uppercase border border-gray-300">Price</th>**

**<th className="px-6 py-3 bg-gray-200 text-gray-700 font-semibold uppercase border border-gray-300">In Stock</th>**

**</tr>**

**</thead>**

**<tbody>**

**{currentShoes.map((shoe, index) => (**

**<tr key={index} className={index % 2 === 0 ? 'bg-gray-100' : ''}>**

**<td className="px-6 py-4 border border-gray-300 text-center">{shoe.shoe\_id}</td>**

**<td className="px-6 py-4 border border-gray-300 text-center">{shoe.brand}</td>**

**<td className="px-6 py-4 border border-gray-300 text-center">{shoe.size}</td>**

**<td className="px-6 py-4 border border-gray-300 text-center">{shoe.color}</td>**

**<td className="px-6 py-4 border border-gray-300 text-center">{shoe.price}</td>**

**<td className="px-6 py-4 border border-gray-300 text-center">{shoe.quantity}</td>**

**</tr>**

**))}**

**</tbody>**

**</table>**

**<div className="flex justify-center mt-4">**

**<button**

**onClick={() => paginate(currentPage - 1)}**

**disabled={currentPage === 1}**

**className="mx-1 px-3 py-1 bg-gray-200 hover:bg-gray-300 rounded-md">**

**Previous**

**</button>**

**<button**

**onClick={() => paginate(currentPage + 1)}**

**disabled={currentShoes.length < shoesPerPage}**

**className="mx-1 px-3 py-1 bg-gray-200 hover:bg-gray-300 rounded-md">**

**Next**

**</button>**

**</div>**

**</div>**

**);**

**};**

**export default Shoetable;**

**ADD SHOES :**

**import React, { useState } from 'react';**

**import { useNavigate } from 'react-router-dom';**

**import axios from 'axios';**

**const AddShoes = () => {**

**const [shoes, setShoes] = useState({**

**shoe\_id: '',**

**brand: '',**

**size: '',**

**color: '',**

**price: '',**

**quantity: ''**

**});**

**const navigate = useNavigate();**

**const handleSubmit = (e) => {**

**e.preventDefault(e);**

**if (!shoes.shoe\_id || !shoes.brand || !shoes.size || !shoes.color || !shoes.price || !shoes.quantity) {**

**alert("All fields are required.");**

**return;**

**}**

**const formData = {**

**shoe\_id : shoes.shoe\_id,**

**brand : shoes.brand,**

**size : shoes.size,**

**color : shoes.color,**

**price : shoes.price,**

**quantity : shoes.quantity,**

**}**

**console.log(formData)**

**axios.post('http://localhost:3000/add\_shoes', formData)**

**.then(result => {**

**if (result.data.Status) {**

**navigate('/shoes');**

**} else {**

**alert(JSON.stringify(result.data));**

**}**

**})**

**.catch(error => {**

**// Handle AxiosError**

**if (error.response) {**

**// Server responded with an error status code**

**console.log('Server responded with:', error.response.status);**

**console.log('Error data:', error.response.data);**

**} else if (error.request) {**

**// Request made but no response received**

**console.log('No response received from the server');**

**} else {**

**// Something else went wrong**

**console.log('Error occurred:', error.message);**

**}**

**});**

**};**

**return (**

**<div className="max-w-md mx-auto shadow-lg ">**

**<div className="p-3 rounded w-96 ">**

**<h3 className="text-xl font-semibold mb-4 mt-2">Add to Shoes Table</h3>**

**<form className="text-md space-y-2" onSubmit={handleSubmit}>**

**<div>**

**<label htmlFor="inputShoeID" className="block">Shoe ID</label>**

**<input**

**type="number"**

**id="inputShoeID"**

**className="border border-gray-300 px-4 py-2 w-full rounded-lg"**

**placeholder="Enter Shoe ID"**

**onChange={(e) => setShoes({ ...shoes, shoe\_id: e.target.value })}**

**/>**

**</div>**

**<div>**

**<label htmlFor="inputBrand" className="block">Brand</label>**

**<input**

**type="text"**

**id="inputBrand"**

**className="border border-gray-300 px-4 py-2 w-full rounded-lg"**

**placeholder="Enter Brand"**

**autoComplete="off"**

**onChange={(e) => setShoes({ ...shoes, brand: e.target.value })}**

**/>**

**</div>**

**<div>**

**<label htmlFor="inputSize" className="block">Size</label>**

**<input**

**type="number"**

**id="inputSize"**

**className="border border-gray-300 px-4 py-2 w-full rounded-lg"**

**placeholder="Enter Size"**

**autoComplete="off"**

**onChange={(e) => setShoes({ ...shoes, size: e.target.value })}**

**min="1"**

**/>**

**</div>**

**<div>**

**<label htmlFor="inputColor" className="block">Color</label>**

**<input**

**type="text"**

**id="inputColor"**

**className="border border-gray-300 px-4 py-2 w-full rounded-lg"**

**placeholder="Enter Color of Shoe"**

**autoComplete="off"**

**onChange={(e) => setShoes({ ...shoes, color: e.target.value })}**

**/>**

**</div>**

**<div>**

**<label htmlFor="inputPrice" className="block">Price</label>**

**<input**

**type="number"**

**id="inputPrice"**

**className="border border-gray-300 px-4 py-2 w-full rounded-lg"**

**placeholder="Enter Price"**

**autoComplete="off"**

**onChange={(e) => setShoes({ ...shoes, price: e.target.value })}**

**min="0"**

**/>**

**</div>**

**<div>**

**<label htmlFor="inputQuantity" className="block">In Stock</label>**

**<input**

**type="number"**

**id="inputQuantity"**

**className="border border-gray-300 px-4 py-2 w-full rounded-lg"**

**placeholder="Enter no. of shoes in stock"**

**autoComplete="off"**

**onChange={(e) => setShoes({ ...shoes, quantity: e.target.value })}**

**min="0"**

**/>**

**</div>**

**<div>**

**<button type="submit" className="bg-blue-500 hover:bg-blue-700 text-white font-bold py-2 px-4 rounded w-full">Add to Table</button>**

**</div>**

**</form>**

**</div>**

**</div>**

**);**

**};**

**export default AddShoes;**

**CUSTOMERS.JSX :**

**import React, {useRef} from 'react'**

**import { Link } from 'react-router-dom'**

**import Customertable from './Customertable'**

**import { useReactToPrint } from 'react-to-print'**

**const Customers = () => {**

**const compRef = useRef(null);**

**const generatePDF = useReactToPrint({**

**content: compRef.current,**

**documentTitle: 'CustomerData',**

**onAfterPrint: ()=> alert("Data saved to pdf")**

**})**

**return (**

**<div className="px-5 w-full h-full flex flex-col justify-center items-center">**

**<div className="text-xl font-bold flex justify-center h-full">**

**<h3 className='text-2xl mt-3 space-y-5'>Customers Table</h3>**

**</div>**

**<button className="hover:bg-gray-800 bg-gray-300 text-black hover:text-white font-bold py-2 px-4 rounded mt-4 inline-block">**

**<Link to="/add\_customers">**

**Add to Table**

**</Link>**

**</button>**

**<Customertable ref={compRef}/>**

**</div>**

**)**

**}**

**export default Customers**

**CUSTOMER TABLE :**

**import React, { useEffect, useState, useRef } from 'react';**

**import axios from 'axios';**

**import { useReactToPrint } from 'react-to-print';**

**const Customertable = () => {**

**const [customers, setCustomers] = useState([]);**

**const [currentPage, setCurrentPage] = useState(1);**

**const [customersPerPage] = useState(5);**

**const componentRef = useRef();**

**const fetchcustomers = async () => {**

**try {**

**const res = await axios.get("http://localhost:3000/customers");**

**setCustomers(res.data.Result);**

**} catch (error) {**

**console.error("Error fetching customers:", error);**

**}**

**};**

**useEffect(() => {**

**fetchcustomers();**

**}, []);**

**// Get current customers for the current page**

**const currentcustomers = customers.slice((currentPage - 1) \* customersPerPage, currentPage \* customersPerPage);**

**// Change page**

**const paginate = (pageNumber) => setCurrentPage(pageNumber);**

**const handlePrint = useReactToPrint({**

**content: () => componentRef.current,**

**documentTitle: 'List\_of\_Customers',**

**});**

**return (**

**<div className="overflow-x-hidden w-full mt-10" ref={componentRef}>**

**<h2 className="text-2xl font-semibold mb-4 text-center">List of Customers</h2>**

**<table className="table-auto w-full border-collapse" >**

**<thead>**

**<tr>**

**<th className="px-6 py-3 bg-gray-200 text-gray-700 font-semibold uppercase border border-gray-300">Customer ID</th>**

**<th className="px-6 py-3 bg-gray-200 text-gray-700 font-semibold uppercase border border-gray-300">Customer Name</th>**

**<th className="px-6 py-3 bg-gray-200 text-gray-700 font-semibold uppercase border border-gray-300">Email</th>**

**<th className="px-6 py-3 bg-gray-200 text-gray-700 font-semibold uppercase border border-gray-300">Phone Number</th>**

**</tr>**

**</thead>**

**<tbody>**

**{currentcustomers.map((customer, index) => (**

**<tr key={index} className={index % 2 === 0 ? 'bg-gray-100' : ''}>**

**<td className="px-6 py-4 border border-gray-300 text-center">{customer.customer\_id}</td>**

**<td className="px-6 py-4 border border-gray-300 text-center">{customer.customer\_name}</td>**

**<td className="px-6 py-4 border border-gray-300 text-center">{customer.email}</td>**

**<td className="px-6 py-4 border border-gray-300 text-center">{customer.phone\_number}</td>**

**</tr>**

**))}**

**</tbody>**

**</table>**

**<div className="flex justify-center mt-4">**

**<button**

**onClick={() => paginate(currentPage - 1)}**

**disabled={currentPage === 1}**

**className="mx-1 px-3 py-1 bg-gray-200 hover:bg-gray-300 rounded-md" >**

**Previous**

**</button>**

**<button**

**onClick={() => paginate(currentPage + 1)}**

**disabled={currentcustomers.length < customersPerPage}**

**className="mx-1 px-3 py-1 bg-gray-200 hover:bg-gray-300 rounded-md" >**

**Next**

**</button>**

**</div>**

**<div className="flex justify-center items-center">**

**<button className='flex justify-center items-center hover:bg-gray-800 bg-gray-300 text-black hover:text-white font-bold py-2 px-4 rounded mt-4' onClick={handlePrint}>**

**Generate PDF**

**</button>**

**</div>**

**</div>**

**);**

**};**

**export default Customertable;**

**ADD CUSTOMERS :**

**import React, { useState } from 'react'**

**import { useNavigate } from 'react-router-dom'**

**import axios from 'axios'**

**const Addcustomers = () => {**

**const [customers, setCustomers] = useState({**

**customer\_id: '',**

**customer\_name:'',**

**email: '',**

**phone\_number:''**

**})**

**const navigate = useNavigate()**

**const handleSubmit = (e) => {**

**e.preventDefault()**

**if (!customers.customer\_id || !customers.customer\_name || !customers.email || !customers.phone\_number) {**

**alert("All fields are required.");**

**return;**

**}**

**const formData = {**

**customer\_id : customers.customer\_id,**

**customer\_name : customers.customer\_name,**

**email : customers.email,**

**phone\_number : customers.phone\_number,**

**}**

**console.log(formData)**

**axios.post('http://localhost:3000/add\_customers', formData)**

**.then(result => {**

**if (result.data.Status) {**

**navigate('/customers');**

**} else {**

**alert(JSON.stringify(result.data));**

**}**

**})**

**.catch(error => {**

**// Handle AxiosError**

**if (error.response) {**

**// Server responded with an error status code**

**console.log('Server responded with:', error.response.status);**

**console.log('Error data:', error.response.data);**

**} else if (error.request) {**

**// Request made but no response received**

**console.log('No response received from the server');**

**} else {**

**// Something else went wrong**

**console.log('Error occurred:', error.message);**

**}**

**});**

**};**

**return (**

**<div className="flex justify-center items-center max-w-md mx-auto shadow-lg ">**

**<div className="p-3 rounded w-96 ">**

**<h3 className="text-xl font-semibold mb-4 mt-2 flex justify-center items-center">Add to Customers Table</h3>**

**<form className="text-md space-y-2" onSubmit={handleSubmit}>**

**<div>**

**<label htmlFor="inputcustomerid" className="block">Customer ID</label>**

**<input**

**type="number"**

**id="inputcustomerid"**

**className="border border-gray-300 px-4 py-2 w-full rounded-lg"**

**placeholder="Enter Customer Id"**

**onChange={(e) => setCustomers({ ...customers, customer\_id: e.target.value })}**

**/>**

**</div>**

**<div>**

**<label htmlFor="inputcustomername" className="block">Customer Name</label>**

**<input**

**type="text"**

**id="inputcustomername"**

**className="border border-gray-300 px-4 py-2 w-full rounded-lg"**

**placeholder="Enter Customer Name"**

**autoComplete="off"**

**onChange={(e) => setCustomers({ ...customers, customer\_name: e.target.value })}**

**/>**

**</div>**

**<div>**

**<label htmlFor="inputemail" className="block">Email</label>**

**<input**

**type="email"**

**id="inputemail"**

**className="border border-gray-300 px-4 py-2 w-full rounded-lg"**

**placeholder="Enter Email"**

**autoComplete="off"**

**onChange={(e) => setCustomers({ ...customers, email: e.target.value })}**

**/>**

**</div>**

**<div>**

**<label htmlFor="inputphonenumber" className="block">Phone Number</label>**

**<input**

**type="number"**

**id="inputphonenumber"**

**className="border border-gray-300 px-4 py-2 w-full rounded-lg"**

**placeholder="Enter Phone Number"**

**autoComplete="off"**

**onChange={(e) => setCustomers({ ...customers, phone\_number: e.target.value })}**

**min="1"**

**/>**

**</div>**

**<div>**

**<button type="submit" className="bg-blue-500 hover:bg-blue-700 text-white font-bold py-2 px-4 rounded w-full">Add to Table</button>**

**</div>**

**</form>**

**</div>**

**</div>**

**);**

**}**

**export default Addcustomers**

**SALES.JSX :**

**import React from 'react'**

**import { Link } from 'react-router-dom'**

**import Salestable from './Salestable'**

**const Sales = () => {**

**return (**

**<div className="px-5 w-full h-full flex flex-col justify-center items-center">**

**<div className="text-xl font-bold flex justify-center h-full">**

**<h3 className='text-2xl mt-3 space-y-5'>Sales Table</h3>**

**</div>**

**<button className="hover:bg-gray-800 bg-gray-300 text-black hover:text-white font-bold py-2 px-4 rounded mt-4 inline-block">**

**<Link to="/add\_sales">**

**Add to Table**

**</Link>**

**</button>**

**<Salestable />**

**</div>**

**)**

**}**

**export default Sales**

**SALES TABLE :**

**import axios from 'axios';**

**import React, { useEffect, useState } from 'react'**

**const Salestable = () => {**

**const [sales, setsales] = useState([]);**

**const [currentPage, setCurrentPage] = useState(1);**

**const [salesPerPage] = useState(5);**

**const fetchsales = async () => {**

**try {**

**const res = await axios.get("http://localhost:3000/sales");**

**setsales(res.data.Result);**

**} catch (error) {**

**console.error("Error fetching sales:", error);**

**}**

**};**

**useEffect(() => {**

**fetchsales();**

**}, []);**

**// Get current sales for the current page**

**const currentsales = sales.slice((currentPage - 1) \* salesPerPage, currentPage \* salesPerPage);**

**// Change page**

**const paginate = (pageNumber) => setCurrentPage(pageNumber);**

**return (**

**<div className="overflow-x-hidden w-full mt-10">**

**<h2 className="text-2xl font-semibold mb-4 text-center">List of Sales</h2>**

**<table className="table-auto w-full border-collapse">**

**<thead>**

**<tr>**

**<th className="px-6 py-3 bg-gray-200 text-gray-700 font-semibold uppercase border border-gray-300">Sale ID</th>**

**<th className="px-6 py-3 bg-gray-200 text-gray-700 font-semibold uppercase border border-gray-300">Purchase ID</th>**

**<th className="px-6 py-3 bg-gray-200 text-gray-700 font-semibold uppercase border border-gray-300">Sale Date</th>**

**<th className="px-6 py-3 bg-gray-200 text-gray-700 font-semibold uppercase border border-gray-300">Sale Amount</th>**

**</tr>**

**</thead>**

**<tbody>**

**{currentsales.map((sale, index) => (**

**<tr key={index} className={index % 2 === 0 ? 'bg-gray-100' : ''}>**

**<td className="px-6 py-4 border border-gray-300 text-center">{sale.sale\_id}</td>**

**<td className="px-6 py-4 border border-gray-300 text-center">{sale.purchase\_id}</td>**

**<td className="px-6 py-4 border border-gray-300 text-center">{sale.sale\_date}</td>**

**<td className="px-6 py-4 border border-gray-300 text-center">{sale.sale\_amount}</td>**

**</tr>**

**))}**

**</tbody>**

**</table>**

**<div className="flex justify-center mt-4">**

**<button**

**onClick={() => paginate(currentPage - 1)}**

**disabled={currentPage === 1}**

**className="mx-1 px-3 py-1 bg-gray-200 hover:bg-gray-300 rounded-md" >**

**Previous**

**</button>**

**<button**

**onClick={() => paginate(currentPage + 1)}**

**disabled={currentsales.length < salesPerPage}**

**className="mx-1 px-3 py-1 bg-gray-200 hover:bg-gray-300 rounded-md" >**

**Next**

**</button>**

**</div>**

**</div>**

**);**

**}**

**export default Salestable**

**ADD SALES INFO :**

**import axios from 'axios'**

**import React from 'react'**

**import { useState } from 'react'**

**import { useNavigate } from 'react-router-dom'**

**const Addsales = () => {**

**const [Sales, setSales] = useState({**

**sale\_id: '',**

**purchase\_id: '',**

**sale\_date: '',**

**sale\_amount: ''**

**})**

**const navigate = useNavigate()**

**const handleSubmit = (e) => {**

**e.preventDefault()**

**if (!Sales.sale\_id || !Sales.purchase\_id || !Sales.sale\_date || !Sales.sale\_amount) {**

**alert("All fields are required.");**

**return;**

**}**

**const formData = {**

**sale\_id: Sales.sale\_id,**

**purchase\_id: Sales.purchase\_id,**

**sale\_date: Sales.sale\_date,**

**sale\_amount: Sales.sale\_amount,**

**}**

**console.log(formData)**

**axios.post('http://localhost:3000/add\_sales', formData)**

**.then(result => {**

**if (result.data.Status) {**

**navigate('/sales');**

**} else {**

**alert(JSON.stringify(result.data));**

**}**

**})**

**.catch(error => {**

**// Handle AxiosError**

**if (error.response) {**

**// Server responded with an error status code**

**console.log('Server responded with:', error.response.status);**

**console.log('Error data:', error.response.data);**

**} else if (error.request) {**

**// Request made but no response received**

**console.log('No response received from the server');**

**} else {**

**// Something else went wrong**

**console.log('Error occurred:', error.message);**

**}**

**});**

**};**

**return (**

**<div className=" max-w-md mx-auto shadow-lg ">**

**<div className="p-3 rounded w-96 ">**

**<h3 className="text-xl font-semibold mb-4 mt-2 flex justify-center items-center">Add to Sales Table</h3>**

**<form className="text-md space-y-2" onSubmit={handleSubmit}>**

**<div>**

**<label htmlFor="inputsaleid" className="block">Sale ID</label>**

**<input**

**type="number"**

**id="inputsaleid"**

**className="border border-gray-300 px-4 py-2 w-full rounded-lg"**

**placeholder="Enter Sale ID"**

**onChange={(e) => setSales({ ...Sales, sale\_id: e.target.value })}**

**/>**

**</div>**

**<div>**

**<label htmlFor="inputpurchaseid" className="block">Purchase ID</label>**

**<input**

**type="number"**

**id="inputpurchaseid"**

**className="border border-gray-300 px-4 py-2 w-full rounded-lg"**

**placeholder="Enter Purchase ID"**

**onChange={(e) => setSales({ ...Sales, purchase\_id: e.target.value })}**

**/>**

**</div>**

**<div>**

**<label htmlFor="inputsaledate" className="block">Sale Date</label>**

**<input**

**type="date"**

**id="inputsaledate"**

**className="border border-gray-300 px-4 py-2 w-full rounded-lg"**

**autoComplete="off"**

**onChange={(e) => setSales({ ...Sales, sale\_date: e.target.value })}**

**/>**

**</div>**

**<div>**

**<label htmlFor="inputsaleamount" className="block">Sale Amount</label>**

**<input**

**type="number"**

**id="inputsaleamount"**

**className="border border-gray-300 px-4 py-2 w-full rounded-lg"**

**placeholder="Enter Sale Amount"**

**autoComplete="off"**

**onChange={(e) => setSales({ ...Sales, sale\_amount: e.target.value })}**

**min="1"**

**/>**

**</div>**

**<div>**

**<button type="submit" className="bg-blue-500 hover:bg-blue-700 text-white font-bold py-2 px-4 rounded w-full">Add to Table</button>**

**</div>**

**</form>**

**</div>**

**</div>**

**);**

**}**

**export default Addsales**

**SUPPLIERS.JSX :**

**import React from 'react'**

**import { Link } from 'react-router-dom'**

**import Supplierstable from './Supplierstable'**

**const Suppliers = () => {**

**return (**

**<div className="px-5 w-full h-full flex flex-col justify-center items-center">**

**<div className="text-xl font-bold flex justify-center h-full">**

**<h3 className='text-2xl mt-3 space-y-5'>Suppliers Table</h3>**

**</div>**

**<button className="hover:bg-gray-800 bg-gray-300 text-black hover:text-white font-bold py-2 px-4 rounded mt-4 inline-block">**

**<Link to="/add\_suppliers">**

**Add to Table**

**</Link>**

**</button>**

**<Supplierstable />**

**</div>**

**)**

**}**

**export default Suppliers**

**SUPPLIERS TABLE :**

**import axios from 'axios';**

**import React, { useEffect, useState } from 'react'**

**const Supplierstable = () => {**

**const [suppliers, setSuppliers] = useState([]);**

**const [currentPage, setCurrentPage] = useState(1);**

**const [suppliersPerPage] = useState(5);**

**const fetchsuppliers = async () => {**

**try {**

**const res = await axios.get("http://localhost:3000/suppliers");**

**setSuppliers(res.data.Result);**

**} catch (error) {**

**console.error("Error fetching suppliers:", error);**

**}**

**};**

**useEffect(() => {**

**fetchsuppliers();**

**}, []);**

**// Get current suppliers for the current page**

**const currentsuppliers = suppliers.slice((currentPage - 1) \* suppliersPerPage, currentPage \* suppliersPerPage);**

**// Change page**

**const paginate = (pageNumber) => setCurrentPage(pageNumber);**

**return (**

**<div className="overflow-x-hidden w-full mt-10">**

**<h2 className="text-2xl font-semibold mb-4 text-center">List of suppliers</h2>**

**<table className="table-auto w-full border-collapse">**

**<thead>**

**<tr>**

**<th className="px-6 py-3 bg-gray-200 text-gray-700 font-semibold uppercase border border-gray-300">Supplier ID</th>**

**<th className="px-6 py-3 bg-gray-200 text-gray-700 font-semibold uppercase border border-gray-300">Supplier Name</th>**

**<th className="px-6 py-3 bg-gray-200 text-gray-700 font-semibold uppercase border border-gray-300">Email</th>**

**<th className="px-6 py-3 bg-gray-200 text-gray-700 font-semibold uppercase border border-gray-300">Phone Number</th>**

**</tr>**

**</thead>**

**<tbody>**

**{currentsuppliers.map((supplier, index) => (**

**<tr key={index} className={index % 2 === 0 ? 'bg-gray-100' : ''}>**

**<td className="px-6 py-4 border border-gray-300 text-center">{supplier.supplier\_id}</td>**

**<td className="px-6 py-4 border border-gray-300 text-center">{supplier.supplier\_name}</td>**

**<td className="px-6 py-4 border border-gray-300 text-center">{supplier.email}</td>**

**<td className="px-6 py-4 border border-gray-300 text-center">{supplier.phone\_number}</td>**

**</tr>**

**))}**

**</tbody>**

**</table>**

**<div className="flex justify-center mt-4">**

**<button**

**onClick={() => paginate(currentPage - 1)}**

**disabled={currentPage === 1}**

**className="mx-1 px-3 py-1 bg-gray-200 hover:bg-gray-300 rounded-md" >**

**Previous**

**</button>**

**<button**

**onClick={() => paginate(currentPage + 1)}**

**disabled={currentsuppliers.length < suppliersPerPage}**

**className="mx-1 px-3 py-1 bg-gray-200 hover:bg-gray-300 rounded-md" >**

**Next**

**</button>**

**</div>**

**</div>**

**);**

**}**

**export default Supplierstable**

**ADD SUPPLIERS :**

**import axios from 'axios'**

**import React from 'react'**

**import { useState } from 'react'**

**import { useNavigate } from 'react-router-dom'**

**const Addsuppliers = () => {**

**const [Suppliers, setSuppliers] = useState({**

**supplier\_id: '',**

**supplier\_name: '',**

**email: '',**

**phone\_number: ''**

**})**

**const navigate = useNavigate()**

**const handleSubmit = (e) => {**

**e.preventDefault()**

**if (!Suppliers.supplier\_id || !Suppliers.supplier\_name|| !Suppliers.email|| !Suppliers.phone\_number) {**

**alert("All fields are required.");**

**return;**

**}**

**const formData = {**

**supplier\_id: Suppliers.supplier\_id,**

**supplier\_name: Suppliers.supplier\_name,**

**email: Suppliers.email,**

**phone\_number: Suppliers.phone\_number,**

**}**

**console.log(formData)**

**axios.post('http://localhost:3000/add\_suppliers', formData)**

**.then(result => {**

**if (result.data.Status) {**

**navigate('/Suppliers');**

**} else {**

**alert(JSON.stringify(result.data));**

**}**

**})**

**.catch(error => {**

**// Handle AxiosError**

**if (error.response) {**

**// Server responded with an error status code**

**console.log('Server responded with:', error.response.status);**

**console.log('Error data:', error.response.data);**

**} else if (error.request) {**

**// Request made but no response received**

**console.log('No response received from the server');**

**} else {**

**// Something else went wrong**

**console.log('Error occurred:', error.message);**

**}**

**});**

**};**

**return (**

**<div className=" max-w-md mx-auto shadow-lg ">**

**<div className="p-3 rounded w-96 ">**

**<h3 className="text-xl font-semibold mb-4 mt-2 flex justify-center items-center">Add to Suppliers Table</h3>**

**<form className="text-md space-y-2" onSubmit={handleSubmit}>**

**<div>**

**<label htmlFor="inputsupplierid" className="block">Supplier ID</label>**

**<input**

**type="number"**

**id="inputsupplierid"**

**className="border border-gray-300 px-4 py-2 w-full rounded-lg"**

**placeholder="Enter Supplier ID"**

**onChange={(e) => setSuppliers({ ...Suppliers, supplier\_id: e.target.value })}**

**/>**

**</div>**

**<div>**

**<label htmlFor="inputsuppliername" className="block">Supplier Name</label>**

**<input**

**type="text"**

**id="inputsuppliername"**

**className="border border-gray-300 px-4 py-2 w-full rounded-lg"**

**placeholder="Enter Supplier Name"**

**onChange={(e) => setSuppliers({ ...Suppliers, supplier\_name: e.target.value })}**

**/>**

**</div>**

**<div>**

**<label htmlFor="inputemail" className="block">Email</label>**

**<input**

**type="email"**

**id="inputemail"**

**className="border border-gray-300 px-4 py-2 w-full rounded-lg"**

**placeholder='Enter Email'**

**autoComplete="off"**

**onChange={(e) => setSuppliers({ ...Suppliers, email: e.target.value })}**

**/>**

**</div>**

**<div>**

**<label htmlFor="inputphonenumber" className="block">Phone Number</label>**

**<input**

**type="number"**

**id="inputphonenumber"**

**className="border border-gray-300 px-4 py-2 w-full rounded-lg"**

**placeholder="Enter Phone Number"**

**autoComplete="off"**

**onChange={(e) => setSuppliers({ ...Suppliers, phone\_number: e.target.value })}**

**min="1"**

**/>**

**</div>**

**<div>**

**<button type="submit" className="bg-blue-500 hover:bg-blue-700 text-white font-bold py-2 px-4 rounded w-full">Add to Table</button>**

**</div>**

**</form>**

**</div>**

**</div>**

**);**

**}**

**export default Addsuppliers**

**PURCHASES.JSX :**

**import React from 'react'**

**import { Link } from 'react-router-dom'**

**import Purchasestable from './Purchasestable'**

**const Purchases = () => {**

**return (**

**<div className="px-5 w-full h-full flex flex-col justify-center items-center">**

**<div className="text-xl font-bold flex justify-center h-full">**

**<h3 className='text-2xl mt-3 space-y-5'>Purchases Table</h3>**

**</div>**

**<button className="hover:bg-gray-800 bg-gray-300 text-black hover:text-white font-bold py-2 px-4 rounded mt-4 inline-block">**

**<Link to="/add\_purchases">**

**Add to Table**

**</Link>**

**</button>**

**<Purchasestable />**

**</div>**

**)**

**}**

**export default Purchases**

**PURCHASES TABLE :**

**import axios from 'axios';**

**import React, { useEffect, useState } from 'react'**

**const Purchasestable = () => {**

**const [purchases, setPurchases] = useState([]);**

**const [currentPage, setCurrentPage] = useState(1);**

**const [purchasesPerPage] = useState(5);**

**const fetchpurchases = async () => {**

**try {**

**const res = await axios.get("http://localhost:3000/purchases");**

**setPurchases(res.data.Result);**

**} catch (error) {**

**console.error("Error fetching purchases:", error);**

**}**

**};**

**useEffect(() => {**

**fetchpurchases();**

**}, []);**

**// Get current purchases for the current page**

**const currentpurchases = purchases.slice((currentPage - 1) \* purchasesPerPage, currentPage \* purchasesPerPage);**

**// Change page**

**const paginate = (pageNumber) => setCurrentPage(pageNumber);**

**return (**

**<div className="overflow-x-hidden w-full mt-10">**

**<h2 className="text-2xl font-semibold mb-4 text-center">List of Purchases</h2>**

**<table className="table-auto w-full border-collapse">**

**<thead>**

**<tr>**

**<th className="px-6 py-3 bg-gray-200 text-gray-700 font-semibold uppercase border border-gray-300">Purchases ID</th>**

**<th className="px-6 py-3 bg-gray-200 text-gray-700 font-semibold uppercase border border-gray-300">Customer ID</th>**

**<th className="px-6 py-3 bg-gray-200 text-gray-700 font-semibold uppercase border border-gray-300">Shoe ID</th>**

**<th className="px-6 py-3 bg-gray-200 text-gray-700 font-semibold uppercase border border-gray-300">Quantity</th>**

**<th className="px-6 py-3 bg-gray-200 text-gray-700 font-semibold uppercase border border-gray-300">Purchase Date</th>**

**</tr>**

**</thead>**

**<tbody>**

**{currentpurchases.map((purchase, index) => (**

**<tr key={index} className={index % 2 === 0 ? 'bg-gray-100' : ''}>**

**<td className="px-6 py-4 border border-gray-300 text-center">{purchase.purchase\_id}</td>**

**<td className="px-6 py-4 border border-gray-300 text-center">{purchase.customer\_id}</td>**

**<td className="px-6 py-4 border border-gray-300 text-center">{purchase.shoe\_id}</td>**

**<td className="px-6 py-4 border border-gray-300 text-center">{purchase.quantity}</td>**

**<td className="px-6 py-4 border border-gray-300 text-center">{purchase.purchase\_date}</td>**

**</tr>**

**))}**

**</tbody>**

**</table>**

**<div className="flex justify-center mt-4">**

**<button**

**onClick={() => paginate(currentPage - 1)}**

**disabled={currentPage === 1}**

**className="mx-1 px-3 py-1 bg-gray-200 hover:bg-gray-300 rounded-md" >**

**Previous**

**</button>**

**<button**

**onClick={() => paginate(currentPage + 1)}**

**disabled={currentpurchases.length < purchasesPerPage}**

**className="mx-1 px-3 py-1 bg-gray-200 hover:bg-gray-300 rounded-md" >**

**Next**

**</button>**

**</div>**

**</div>**

**);**

**}**

**export default Purchasestable**

**ADD PURCHASES :**

**import axios from 'axios'**

**import React from 'react'**

**import { useState } from 'react'**

**import { useNavigate } from 'react-router-dom'**

**const Addpurchases = () => {**

**const [purchases, setPurchases] = useState({**

**purchase\_id: '',**

**customer\_id: '',**

**shoe\_id: '',**

**quantity: '',**

**purchase\_date: '',**

**})**

**const navigate = useNavigate()**

**const handleSubmit = (e) => {**

**e.preventDefault()**

**if (!purchases.purchase\_id || !purchases.customer\_id|| !purchases.shoe\_id|| !purchases.quantity || !purchases.purchase\_date) {**

**alert("All fields are required.");**

**return;**

**}**

**const formData = {**

**purchase\_id: purchases.purchase\_id,**

**customer\_id: purchases.customer\_id,**

**shoe\_id: purchases.shoe\_id,**

**quantity: purchases.quantity,**

**purchase\_date: purchases.purchase\_date,**

**}**

**console.log(formData)**

**axios.post('http://localhost:3000/add\_purchases', formData)**

**.then(result => {**

**if (result.data.Status) {**

**navigate('/purchases');**

**} else {**

**alert(JSON.stringify(result.data));**

**}**

**})**

**.catch(error => {**

**// Handle AxiosError**

**if (error.response) {**

**// Server responded with an error status code**

**console.log('Server responded with:', error.response.status);**

**console.log('Error data:', error.response.data);**

**} else if (error.request) {**

**// Request made but no response received**

**console.log('No response received from the server');**

**} else {**

**// Something else went wrong**

**console.log('Error occurred:', error.message);**

**}**

**});**

**};**

**return (**

**<div className=" max-w-md mx-auto shadow-lg ">**

**<div className="p-3 rounded w-96 ">**

**<h3 className="text-xl font-semibold mb-4 mt-2 flex justify-center items-center">Add to Purchases Table</h3>**

**<form className="text-md space-y-2" onSubmit={handleSubmit}>**

**<div>**

**<label htmlFor="inputpurchaseid" className="block">Purchase ID</label>**

**<input**

**type="number"**

**id="inputpurchaseid"**

**className="border border-gray-300 px-4 py-2 w-full rounded-lg"**

**placeholder="Enter Purchase ID"**

**onChange={(e) => setPurchases({ ...purchases, purchase\_id: e.target.value })}**

**/>**

**</div>**

**<div>**

**<label htmlFor="inputcustomerid" className="block">Customer ID</label>**

**<input**

**type="number"**

**id="inputcustomerid"**

**className="border border-gray-300 px-4 py-2 w-full rounded-lg"**

**placeholder="Enter Customer ID"**

**onChange={(e) => setPurchases({ ...purchases, customer\_id: e.target.value })}**

**/>**

**</div>**

**<div>**

**<label htmlFor="inputshoeid" className="block">Shoe ID</label>**

**<input**

**type="number"**

**id="inputshoeid"**

**className="border border-gray-300 px-4 py-2 w-full rounded-lg"**

**placeholder="Enter Shoe ID"**

**onChange={(e) => setPurchases({ ...purchases, shoe\_id: e.target.value })}**

**/>**

**</div>**

**<div>**

**<label htmlFor="inputquantity" className="block">Quantity</label>**

**<input**

**type="number"**

**id="inputquantity"**

**className="border border-gray-300 px-4 py-2 w-full rounded-lg"**

**placeholder='Enter Quantity'**

**autoComplete="off"**

**onChange={(e) => setPurchases({ ...purchases, quantity: e.target.value })}**

**min="1"**

**/>**

**</div>**

**<div>**

**<label htmlFor="inputpurchasedate" className="block">Purchase Date</label>**

**<input**

**type="date"**

**id="inputpurchasedate"**

**className="border border-gray-300 px-4 py-2 w-full rounded-lg"**

**placeholder="Enter Purchase Date"**

**autoComplete="off"**

**onChange={(e) => setPurchases({ ...purchases, purchase\_date: e.target.value })}**

**/>**

**</div>**

**<div>**

**<button type="submit" className="bg-blue-500 hover:bg-blue-700 text-white font-bold py-2 px-4 rounded w-full">Add to Table</button>**

**</div>**

**</form>**

**</div>**

**</div>**

**);**

**}**

**export default Addpurchases**